

UNITED STATES DISTRICT COURT  
EASTERN DISTRICT OF VIRGINIA  
ALEXANDRIA DIVISION

---

CELLEBRITE MOBILE  
SYNCHRONIZATION LTD., and  
CELLEBRITE USA, INC.

Plaintiffs,

v.

MICRO SYSTEMATION AB, and  
MSAB, INC.,

Defendants.

---

Civil Action No. 13-CV-1014 (TSE/TRJ)

FOURTH EXPERT DECLARATION OF  
ROBERT ZEIDMAN

DECLARATION IN SUPPORT OF PLAINTIFFS' MOTION TO COMPEL SOURCE  
CODE DISCOVERY FROM DEFENDANTS

**Table of Contents**

I. Relevant Background..... 1

II. Significance of Source Code Repository and Source Code Versions..... 1

III. Specific Versions of Source Code ..... 3

IV. Significance of Ancillary Files ..... 4

V. Specific Files..... 5

VI. Irrelevance of Source Code Unrelated to Alleged Infringement ..... 5

VII. Conclusion ..... 6

I, Robert Zeidman, provide the following expert disclosures.

**I. RELEVANT BACKGROUND**

1. I have personally consulted on over 125 intellectual property litigations and have written a book entitled *The Software IP Detective's Handbook: Measurement, Comparison, and Infringement Detection*, considered by many to be the seminar book in the field of software forensics. I have created the CodeSuite® set of software tools for comparing software code to find IP infringement, and I train other experts on these tools and methodologies for examining software code to find infringement. These tools, and the methodologies I have developed, allow a trained forensic software analyst to examine many megabytes of source code and object code in only a fraction of the time it would otherwise take. My team and I have exhaustively examined hundreds of megabytes of code, in over 65 cases, for evidence of copying and misappropriation in timeframes measured in days, even though other experts in the field had claimed it was an impossible task. We have found evidence of copying and misappropriation that other experts have missed, and we have also shown the absence of copying and misappropriation when other teams of experts could not come to a definitive conclusion.

**II. SIGNIFICANCE OF SOURCE CODE REPOSITORY AND SOURCE CODE VERSIONS**

2. It is very important in an exhaustive analysis of infringement to examine all version of relevant code going back some time prior to the alleged infringement. Normal software development involves much writing of code, testing, debugging, and modifying. Traces of infringement would be most apparent at the time the infringing code was integrated into an entire software project. After that time, normal software development would result in changes to the code that would make the infringement much more difficult to detect. Furthermore, if a party attempts to purposely hide the infringement, the earliest version of

code would be closer to the original infringed code than later version of the code where the infringer would take active steps to modify the code in ways that did not significantly change its functionality but that cosmetically changed its appearance. For this reason, it is critical to a conclusive infringement analysis that all versions of the relevant development source code be turned over for examination.

3. Most companies that develop software maintain their code in a source code repository where changes to the code are recorded and can be undone if necessary. In other words, if a programmer rewrites a section of code that introduces a fatal flaw, that programmer can "back out" or reverse those changes by extracting a version of the code from the date before those changes were entered. Also, if it is unclear whose changes created the flaw, the repository can be queried to see which programmer entered the code that caused the flaw.
4. Repositories are critical for an exhaustive infringement analysis because it will allow earlier versions of code to be examined, which will be more likely to show infringing code if it exists. The repository will also show anomalous entries of code. For example, code is typically entered a little at a time by a programmer who is developing the code. A large, sudden entry of code can be a sign that the code was developed outside the company and entered into the system *in toto*. These logs can also support an historical understanding of infringement by showing which person entered infringing software code into the system, and when that person entered the infringing software code.
5. I understand that MSAB has claimed that the production of their source code repository would be unduly burdensome because it is very large and would require significant time to transfer the repository to a very large disk. I do not understand this argument. First, the source code repository is already stored somewhere at MSAB on a disk. Sending us a copy would only require copying that repository onto a new disk. My team and I have worked on some of the largest code bases in existence. While that code base may consist of 100 Mbytes of source code, and there may be 1,000 versions of that code, that would require only a 100 Gigabyte drive, which is readily available at the local computer store. Even if

my estimate were off by a factor of 10, this code would easily fit on a 1 Terabyte drive, which is also available at most local computer stores. However, version control systems do not store each line of code from each version of software in the repository. One advantage of a version control system is that it only stores the changes from one version to the next. Thus, a program consisting of 1,000 versions of 100 Megabytes of source code would be unlikely to contain more than 200 Megabytes total, which would easily fit onto the smallest disk available at the local computer store.

### III. SPECIFIC VERSIONS OF SOURCE CODE

6. It is typical in my investigations to get source code repositories for examination because they contains all the code and metadata that could be relevant to my investigation. Only if the company did not use a version control system, would I instead request multiple versions of the relevant source code. In that case, I would typically request source code from dates at least shortly before the alleged infringement, shortly after the alleged infringement, the first released version after the alleged infringement, and the latest released version.
7. I understand that in this case there is a possibility that instead of receiving the source code repository, I may receive multiple versions of source code to examine. While this is not ideal, if this is the case then I need to receive at least the following versions of source code:
  - 1) A version of MSAB source code shortly before March 2011, the date of the release of Samsung Physical by Cellebrite in UFED 2.0.
  - 2) A version of MSAB source code roughly two months after March 2011, the date of the release of Samsung Physical by Cellebrite in UFED 2.0.
  - 3) A version of MSAB source code shortly before January 2012, the date of the release of BlackBerry Physical by Cellebrite UFED ver. 1.1.9.0.
  - 4) A version of MSAB source code roughly two months after January 2012, the date of the release of BlackBerry Physical by Cellebrite UFED ver. 1.1.9.0.
  - 5) A version of MSAB source code shortly before May 2012, the date of the release

of Cellebrite UFED ver. 1197 for data extraction of Samsung Android devices.

- 6) A version of MSAB source code roughly two months after May 2012, the date of the release of Cellebrite UFED ver. 1197 for data extraction of Samsung Android devices.
- 7) The release version of MSAB source code from May 2012, the date of the release of XRY version 6.3 containing Samsung Physical by MSAB.
- 8) The release version of MSAB source code from November 2012, the date of the release of XRY version 6.4.1 containing BlackBerry Physical by MSAB.
- 9) The release version of MSAB source from June 2013, the date of the release of MSAB XRY 6.6 software containing Samsung Android solution.
- 10) The release version of MSAB source code from September 2013, the date of the release MSAB XRY 6.7 adding three additional Samsung Android Models.
- 11) The latest release version of MSAB source code.

#### **IV. SIGNIFICANCE OF ANCILLARY FILES**

8. Ancillary files such as project files, make files, header files, and configuration files provide information about how the software is built and indicate whether all of the files have been provided. In my experience there have been times when some critical files containing infringing code have been left out, whether inadvertently or purposely, and I could only discover this by examining the ancillary files that reference files that were missing from the software code production. It is typical in my investigations to get these ancillary files in the source code repository. Only if the company did not use a version control system, would I instead obtain multiple versions of the relevant ancillary files from dates at least immediately before the alleged infringement, immediately after the alleged infringement, the first released version after the alleged infringement, and the latest released version.

**V. SPECIFIC FILES**

9. In particular, the source code files and ancillary files that I need from MSAB, whether they are produced in a repository or whether I receive explicit versions, are those that specifically relate to their solutions for Samsung, BlackBerry, and Samsung Android. Because these are the files that contain code that allegedly contain misappropriated code, these are the only files I need to examine.

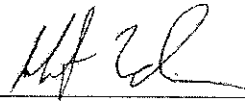
**VI. IRRELEVANCE OF SOURCE CODE UNRELATED TO ALLEGED INFRINGEMENT**

10. In my opinion, it is completely unnecessary for MSAB's experts to see software source code that is not at issue in this litigation. It is my understanding that the percentage of lines of code is irrelevant to any infringement case. Copyright and trade secret infringement rests on the significance of the code, not the number of lines of code. The number of lines of code is independent from the significance. A small routine that has important functionality or is expressed in a particularly creative way can be critical to a company and important to protect even though the number of lines of code may be small.
11. The entire UFED source code sought by Defendants contains over 13 years of proprietary developments, which contains the code for UFED, as well as the code for Cellebrite's UME consumer products, which are not at issue in this case. I understand that the Plaintiffs have already disclosed the source code for the Samsung Solution and the BlackBerry Solution, which are the basis for Plaintiffs' claims of copying against the Defendants. I have been told that Plaintiffs will also produce the source code for the Samsung Android Solution and the Windows Mobile Solution, which relate to Plaintiffs' Amended Complaint and Defendants' counterclaims.
12. Thus, I believe that MSAB's request for the entire UFED code is unwarranted given that most of that code is not at issue in this case.

## VII. CONCLUSION

13. I believe that MSAB should be required to turn over its source code repository that comes from its version control system and that includes all of the relevant software code and ancillary files that are at issue in this case. If MSAB does not utilize a version control system, then MSAB should be required to turn over specific versions of the relevant software source code and ancillary files as requested by Cellebrite. I believe that Cellebrite should not be required to turn over software code or other files that are not relevant to the issues in this case.

Dated: November 26, 2013



Robert Zeidman